Review Problem 12

- How would the ALU be used to help with each of the following branches? The first is filled in for you:
  - B.EQ: SUBS X31, <val1>, <val2>; use zero flag
  - B.NE: (Blank)
  - B.GE: (Blank)
  - B.GT: (Blank)
  - B.LE: (Blank)
  - B.LT: (Blank)
Computer "Performance"

Readings: 1.6-1.8

BIPS (Billion Instructions Per Second) vs. GHz (Giga Cycles Per Second)

Throughput (jobs/seconds) vs. Latency (time to complete a job)

Measuring "best" in a computer

3.0 GHz

The PowerBook G4 outguns Pentium III-based notebooks by up to 30 percent.*

* Based on Adobe Photoshop tests comparing a 500MHz PowerBook G4 to 850MHz Pentium III-based portable computers

Hyper Pipelined Technology
Performance Example: Homebuilders

<table>
<thead>
<tr>
<th>Builder</th>
<th>Time per House</th>
<th>Houses Per Month</th>
<th>House Options</th>
<th>Dollars Per House</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-build</td>
<td>24 months</td>
<td>1/24</td>
<td>Infinite</td>
<td>$200,000</td>
</tr>
<tr>
<td>Contractor</td>
<td>3 months</td>
<td>1</td>
<td>100</td>
<td>$400,000</td>
</tr>
<tr>
<td>Prefab</td>
<td>6 months</td>
<td>1,000</td>
<td>1</td>
<td>$250,000</td>
</tr>
</tbody>
</table>

Which is the "best" home builder?
- Homeowner on a budget?
- Rebuilding Haiti?
- Moving to wilds of Alaska?

Dollars per house: Self-build
Houses per month: Prefab
Time to house: Contractor

Which is the "speediest" builder?
- Latency: how fast is one house built?
- Throughput: how long will it take to build a large number of houses?

CPUs
Computer Performance

Primary goal: execution time (time from program start to program completion)

\[ \text{Performance} = \frac{1}{\text{Execution Time}} \]

To compare machines, we say “X is n times faster than Y”

\[ n = \frac{\text{Performance}_x}{\text{Performance}_y} = \frac{\text{Execution Time}_y}{\text{Execution Time}_x} \]

Example: Machine Orange and Grape run a program
   Orange takes 5 seconds, Grape takes 10 seconds

\[ n = \frac{\text{Perf Orange}}{\text{Perf grape}} = \frac{\text{Exec Grape}}{\text{Exec Orange}} = \frac{10}{5} = 2 \]

Orange is \( 2 \times \) times faster than Grape
Execution Time

Elapsed Time

counts everything \textit{(disk and memory accesses, I/O, etc.)}
a useful number, but often not good for comparison purposes

CPU time

doesn't count I/O or time spent running other programs

can be broken up into system time, and user time

Example: Unix "time" command

\texttt{linux15.ee.washington.edu}\texttt{> time javac CircuitViewer.java}

\begin{center}
3.370u 0.570s 0:12.44 31.6%
\end{center}

\underline{User} \underline{System} \underline{Elapsed}

Our focus: user CPU time

time spent executing the lines of code that are "in" our program
CPU Time

\[ 1 GHz = 1 \times 10^9 \]
\[ 1\mu s \]

CPU execution time for a program = CPU clock cycles for a program * Clock period

CPU execution time for a program = CPU clock cycles for a program * \( \frac{1}{\text{Clock rate}} \)

Application example:

A program takes 10 seconds on computer Orange, with a 400MHz clock. Our design team is developing a machine Grape with a much higher clock rate, but it will require 1.2 times as many clock cycles. If we want to be able to run the program in 6 seconds, how fast must the clock rate be?

\[ 10 \text{ seconds} = \text{Clock cycles}_{\text{Orange}} \times \frac{1}{0.4 \times 10^9} \]

\[ 6 \text{ seconds} = 1.2 \times (4 \times 10^9) \times \frac{1}{\text{Rate}} \]

\[ \text{Rate} = \frac{1.2 \times (4 \times 10^9)}{6} = 0.2 \times 4 \times 10^9 = 0.8 \times 10^9 \]

800 MHz
CPI

How do the # of instructions in a program relate to the execution time?

\[
\text{CPU clock cycles for a program} = \text{Instructions for a program} \times \text{Average Clock Cycles per Instruction (CPI)}
\]

\[
\text{CPU execution time for a program} = \text{Instructions for a program} \times \text{CPI} \times \frac{1}{\text{Clock rate}}
\]
CPI Example

Suppose we have two implementations of the same instruction set (ISA).

For some program

Machine A has a clock cycle time of 10 ns. and a CPI of 2.0
Machine B has a clock cycle time of 20 ns. and a CPI of 1.2

What machine is faster for this program, and by how much?

\[
\frac{Per_A}{Per_B} = \frac{Exec_B}{Exec_A} = \frac{Instr_B \times CPI_B \times Period_B}{Instr_A \times CPI_A \times Period_A} = \frac{1.2 \times 20}{2.0 \times 10} = \frac{2.4}{2.0} = 1.2
\]

A is 1.2x faster than B
Computing CPI

Different types of instructions can take very different amounts of cycles. Memory accesses, integer math, floating point, control flow

\[ CPI = \sum_{\text{types}} \left( \text{Cycles}_{\text{type}} \times \text{Frequency}_{\text{type}} \right) \]

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Type Cycles</th>
<th>Type Frequency</th>
<th>Cycles * Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>1</td>
<td>x</td>
<td>0.5</td>
</tr>
<tr>
<td>Load</td>
<td>5</td>
<td>x</td>
<td>1.0</td>
</tr>
<tr>
<td>Store</td>
<td>3</td>
<td>x</td>
<td>0.3</td>
</tr>
<tr>
<td>Branch</td>
<td>2</td>
<td>x</td>
<td>0.4</td>
</tr>
</tbody>
</table>

CPI: 2.2